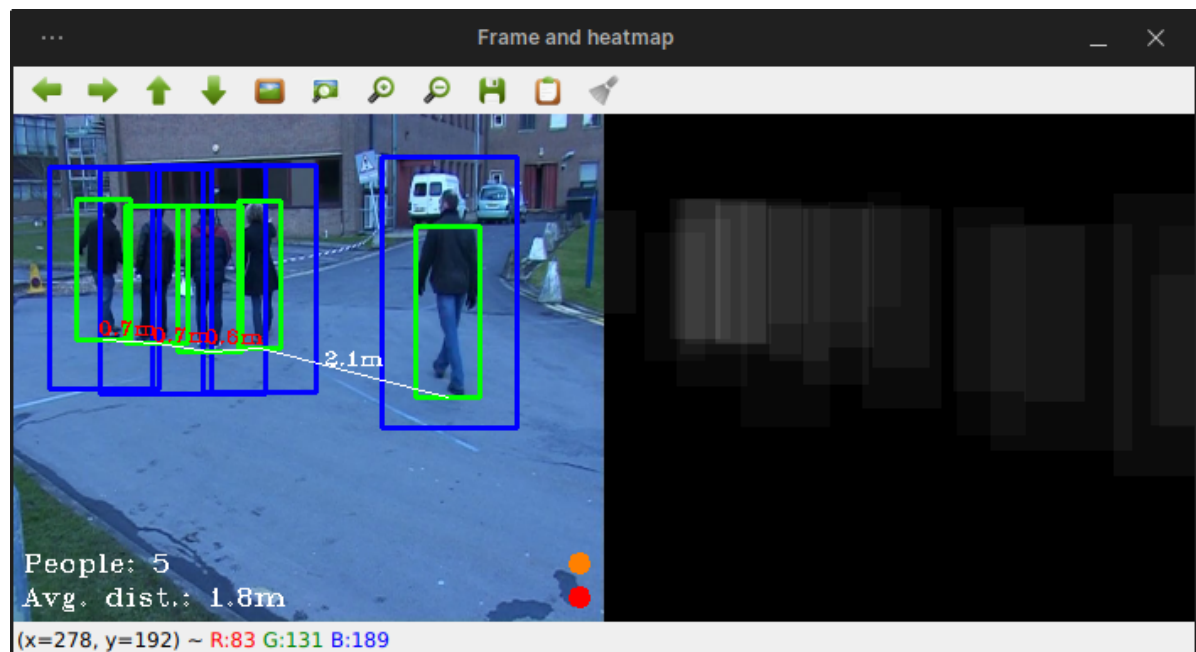


# HumanCount

This repository contains the final project's code for the **Computer Vision** course at Sapienza University of Rome (A.Y. 2021-22).



The project does **real-time human counting** using HOG and SVM, and it is intended to be used to **monitor a known area**. In addition to counting, the program:

- Estimates the **distance** between people.
- Outputs visual and terminal **alarms** when:
  - The number of people is higher than the maximum allowed.
  - The minimum distance between people is less than the minimum allowed.
- Shows an evolving **heat-map**, to describe the density of the people over certain areas, over time.

In order to give better distance estimations, we needed to improve the size of the bounding boxes. To reach this goal, the approaches described in the paper **Human Detection Using HOG-SVM, Mixture of Gaussian and Background Contours Subtraction** by A. H. Ahmed, K. Kpalma and A. O. Guedi has been used as a reference.

A. H. Ahmed, K. Kpalma and A. O. Guedi, "[Human Detection Using HOG-SVM, Mixture of Gaussian and Background Contours Subtraction](#)," 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2017, pp. 334-338, doi: 10.1109/SITIS.2017.62.

## Prerequisites

The only requirements to run the project are:

- OpenCV 4.5.3+
- NumPy

## Running

The entry-point is the `main.py` script, which can be configured via multiple command line arguments; specifically:

```

1  usage: main.py [-h] -i INPUT [-sh] [-nf] [-m] [-nac] [-nad] [-s]
2
3  optional arguments:
4    -h, --help            show this help message and exit
5    -i INPUT, --input INPUT
6                          Input JSON
7    -sh, --show-hog-boxes
8                          Show the bounding boxes produced by HOG-SVM
9    -nf, --no-filter-optimized-boxes
10                         Also show the optimized bounding boxes that are
11                         outside the HOG-SVM ones
12    -m, --use-mog2        Use MOG2 to perform background subtraction
13    -nac, --no-alarm-count
14                         Disable the alarm when the number of people
15                         exceeds the limit
16    -nad, --no-alarm-distance
17                         Disable the alarm when there is a distance smaller
18                         than the minimum limit allowed
19    -s, --show            Show some intermediate preprocessing steps in a
20                         window

```

By default, the script looks for the still `background` image provided in the input JSON; alternatively, if the `--use-mog2` option is provided, the background will be extracted in real-time for each frame.

The background image and the video to use must be in the **same directory** as the JSON file.

## JSON configuration

The main script needs a JSON file as input, containing the configuration of the video/feed to use, for example:

```

1  {
2    "video": "5.mp4",
3    "background": "5.png",
4    "camera_conf": {
5      "height": 2.0,
6      "lower_angle": 55,
7      "upper_angle": 100
8    },
9    "alarms": {
10     "max_people": 4,
11     "min_distance": 1
12   }
13 }

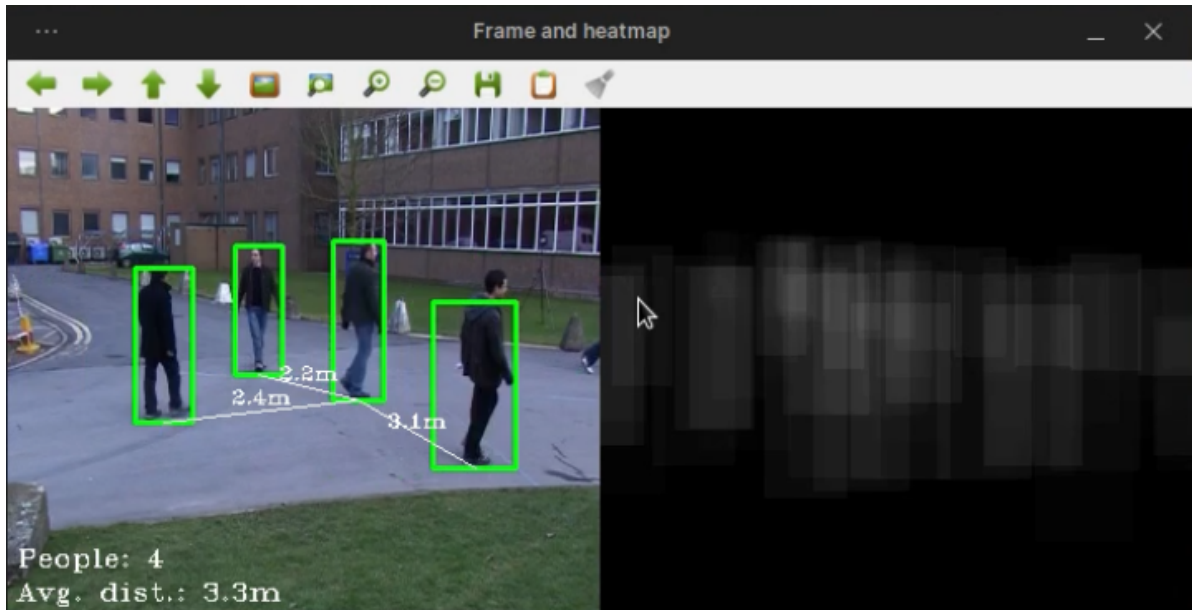
```

- If not available, the background image can be estimated from an already made video, by using the `background_estimator.py` script.
- The `camera_conf` section contains the known values of the camera, that are used to estimate the distance between the camera and the people.
- The `alarms` section contains parameters/rules to decide when to raise an alarm state.

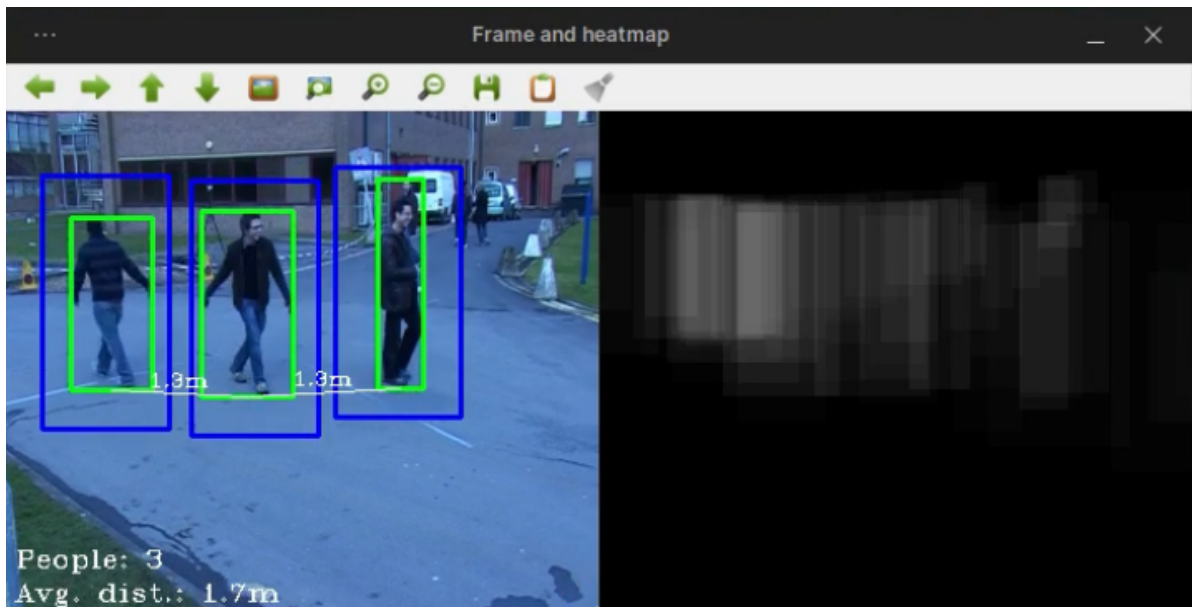
## Examples

This section contains some example, explaining the outcome of certain command line arguments.

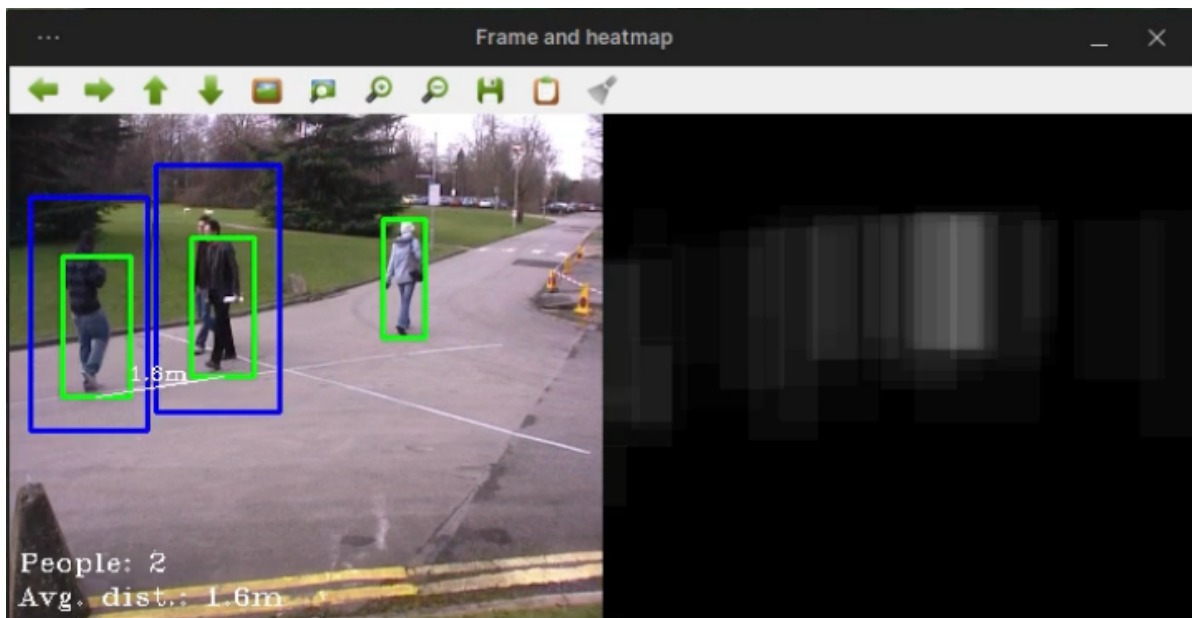
```
1 # show the filtered optimized boxes (default)
2 python3 main.py -i datasets/5.json
```



```
1 # also show the hog-generated bounding boxes (blue)
2 python3 main.py -i datasets/6.json --show-hog-boxes
```



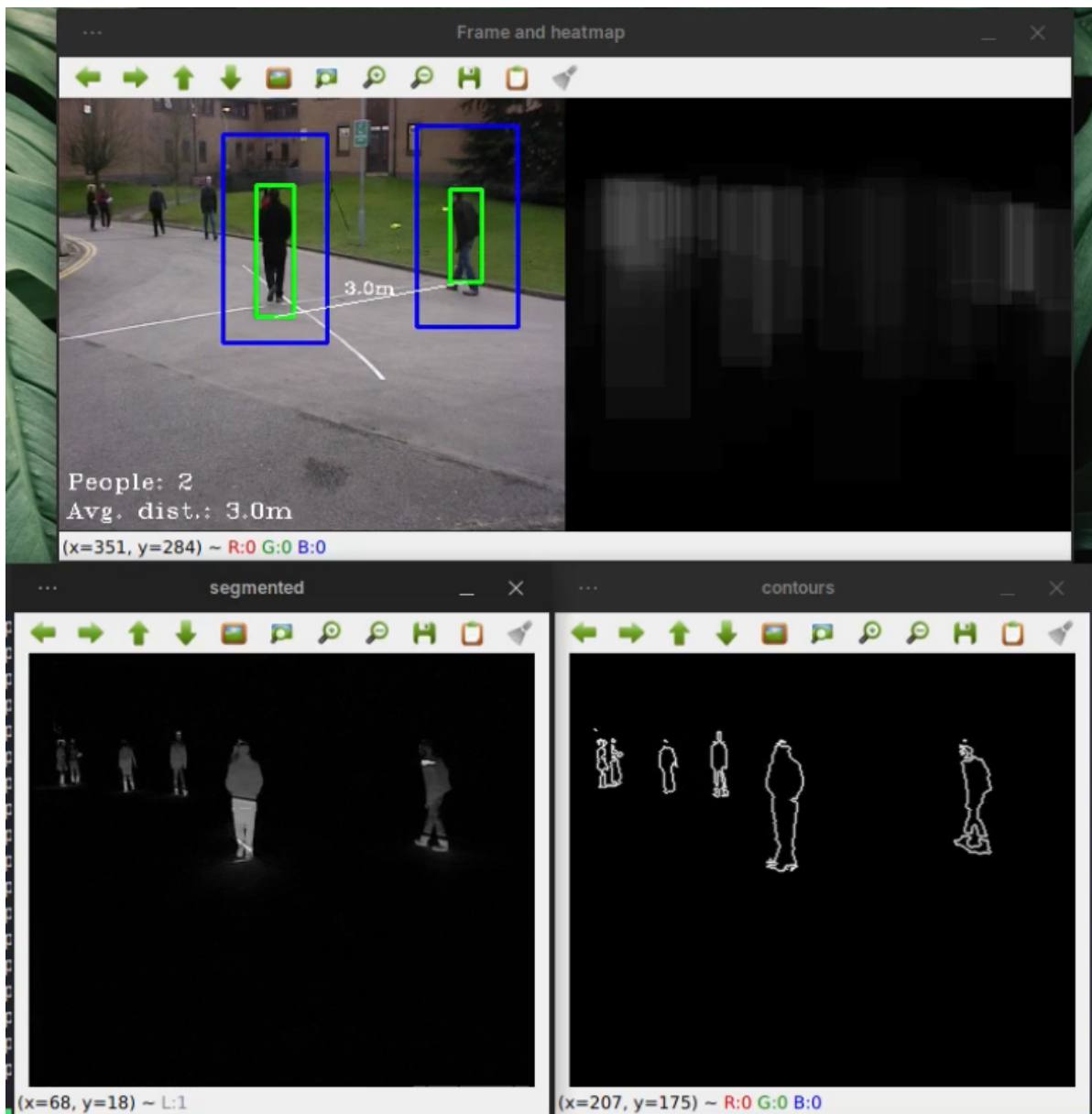
```
1 # also show the hog-generated bounding boxes (blue), and show the optimized
  ones even if they are not in an hog box (no filter)
2 python3 main.py -i datasets/7.json --show-hog-boxes --no-filter-optimized-
  boxes
```



```

1 # also show the hog-generated bounding boxes (blue),
2 # and some preprocessing steps in separate windows (segmentation, countours
  extraction)
3 python3 main.py -i datasets/8.json --show-hog-boxes --show

```



## Terminal output

In addition to the visual signals (colored circles in the right bottom, and colored distances), alarms are logged to `stdout`.

```
*** Terminale -
[People count alarm] Current: 7;          maximum allowed: 4
[People distance alarm] Found: 0.64;      minimum allowed: 1
[People count alarm] Current: 6;          maximum allowed: 4
[People count alarm] Current: 6;          maximum allowed: 4
[People count alarm] Current: 6;          maximum allowed: 4
[People count alarm] Current: 6;          maximum allowed: 4
[People count alarm] Current: 5;          maximum allowed: 4
[People count alarm] Current: 5;          maximum allowed: 4
[People count alarm] Current: 5;          maximum allowed: 4
[People count alarm] Current: 5;          maximum allowed: 4
[People count alarm] Current: 5;          maximum allowed: 4
[People distance alarm] Found: 0.58;      minimum allowed: 1
[People distance alarm] Found: 0.72;      minimum allowed: 1
[People count alarm] Current: 5;          maximum allowed: 4
[People distance alarm] Found: 0.76;      minimum allowed: 1
[People count alarm] Current: 6;          maximum allowed: 4
[People distance alarm] Found: 0.8;        minimum allowed: 1
[People count alarm] Current: 5;          maximum allowed: 4
```